Examining and Supporting Multi-Tasking in EV3OSEK

Nils Hölscher, Kuan-Hsun Chen,
Georg von der Brüggen, and Jian-Jia Chen

Department of Computer Science, Chair 12
TU Dortmund, Germany

3rd of July 2018

This research is supported by DFG, as part of the Collaborative Research Center SFB876 (http://sfb876.tu-dortmund.de/), subproject A1.

Outline

Introduction of EV3OSEK

Application Model

Issues in EV3OSEK

Proposed Solutions

Evaluation

Current State of EV3OSEK

EV3OSEK

- EV3OSEK is an OS for Lego Mindstorms EV3 (2013).
    - aims to fulfil the OSEK standard.
    - NXTOSEK port by Westsächsische Hochschule Zwickau.

EV3OSEK

- EV3OSEK is an OS for Lego Mindstorms EV3 (2013).
  - aims to fulfil the OSEK standard.
  - NXTOSEK port by Westsächsische Hochschule Zwickau.

- SoC TexsInstruments AM1808

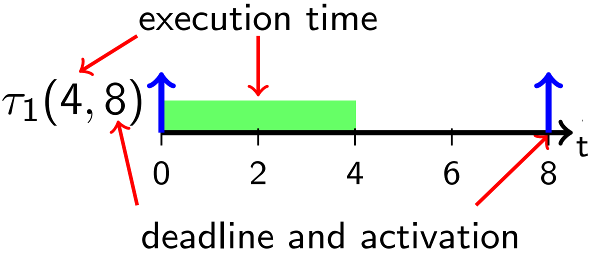
EV3OSEK

- EV3OSEK is an OS for Lego Mindstorms EV3 (2013).
  - aims to fulfil the OSEK standard.
  - NXTOSEK port by Westsächsische Hochschule Zwickau.

- SoC TexsInstruments AM1808
  - ARM926EJ-S
  - ARM9
  - 300MHz
  - 64 MB RAM

Application Model

- Priority decreases with higher indices: $p(\tau_{i-1}) > p(\tau_i) > p(\tau_{i+1})$.
- Deadlines are implicit: $T_i = D_i$
- Job misses deadline: related release is skipped.



NXTOSEK and EV3OSEK

- NXTOSEK is an OS for Lego Mindstorms NXT (2006).
  - uses Toppers/JSP or Toppers/ATK(OSEK) kernel.
  - has to be flashed on the brick.

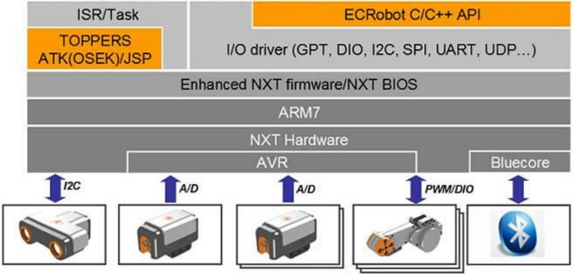


Figure: http://lejos-osek.sourceforge.net/

NXTOSEK and EV3OSEK

- NXTOSEK is an OS for Lego Mindstorms NXT (2006).
  - uses Toppers/JSP or Toppers/ATK(OSEK) kernel.
  - has to be flashed on the brick.
- Only the Toppers ATK(OSEK) kernel has been ported to EV3.
- ECRobot API in EV3OSEK supports less hardware in EV3.

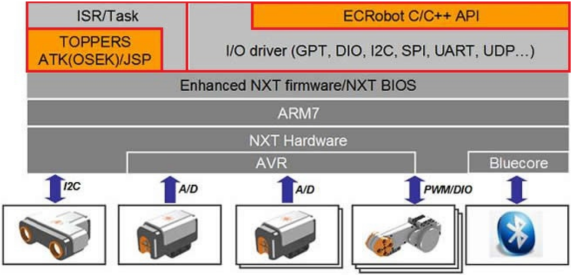
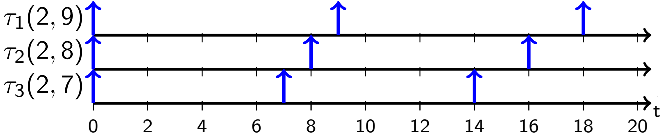


Figure: http://lejos-osek.sourceforge.net/

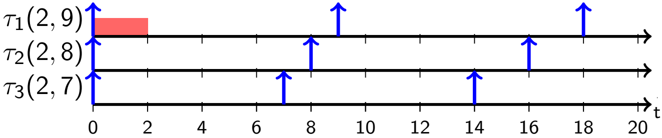
Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.



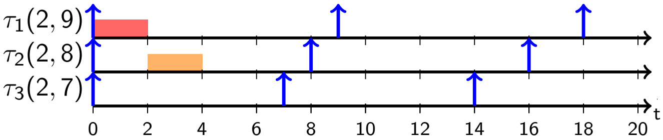
Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.



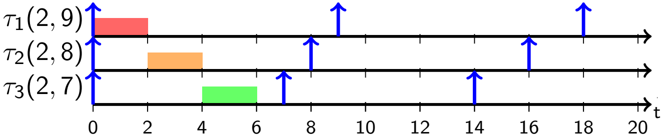
Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.



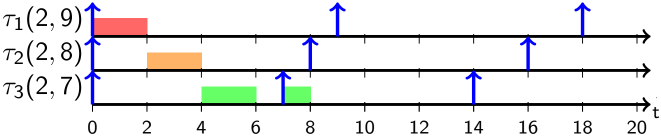
Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.



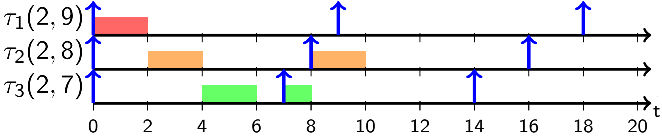
Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.



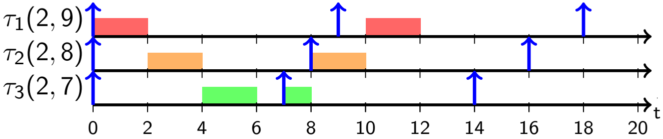
Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.



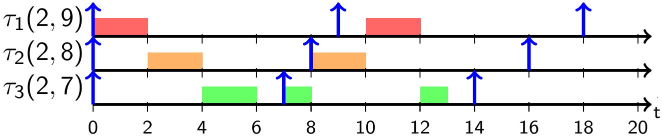
Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.



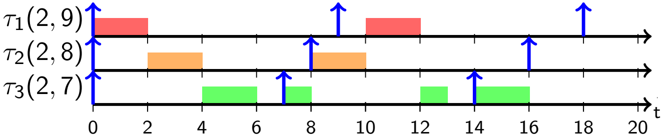
Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.



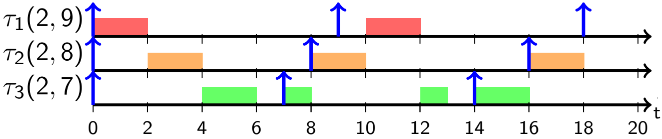
Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.



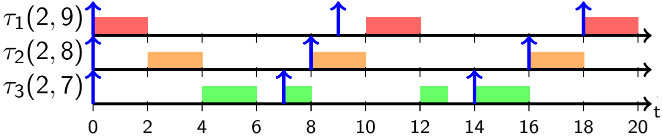
Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.



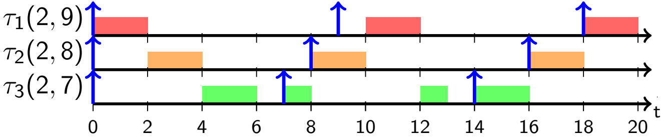
Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.

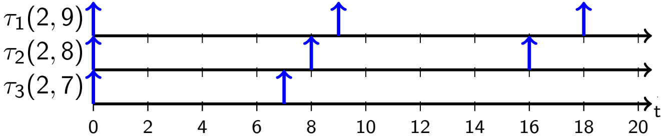


Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.

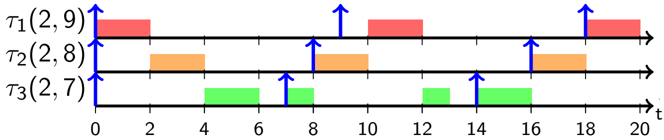


- Behaviour of Guptas and Doshis implementation of nested preemption in NXTOSEK (technical report in 2014).

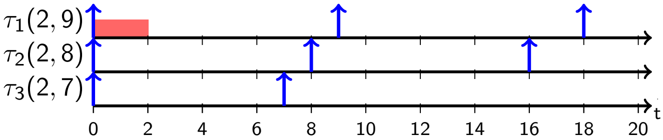


Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.

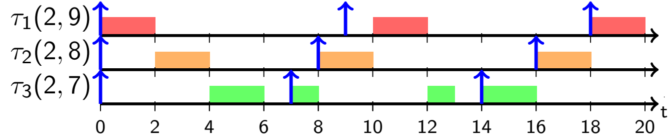


- Behaviour of Guptas and Doshis implementation of nested preemption in NXTOSEK (technical report in 2014).

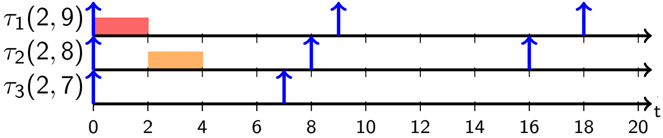


Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.

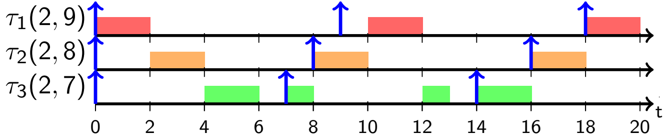


- Behaviour of Guptas and Doshis implementation of nested preemption in NXTOSEK (technical report in 2014).

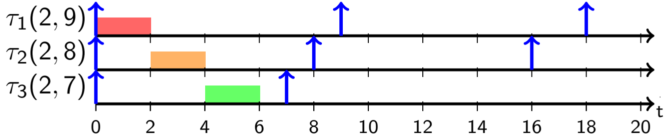


Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.

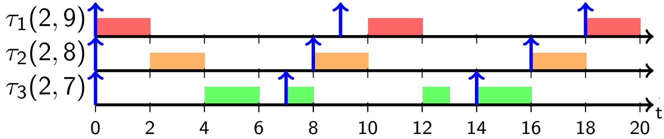


- Behaviour of Guptas and Doshis implementation of nested preemption in NXTOSEK (technical report in 2014).

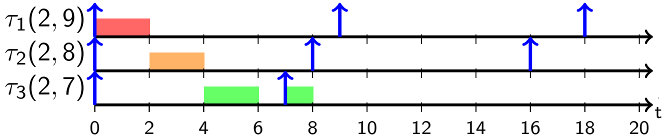


Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.

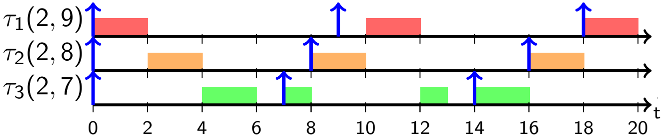


- Behaviour of Guptas and Doshis implementation of nested preemption in NXTOSEK (technical report in 2014).

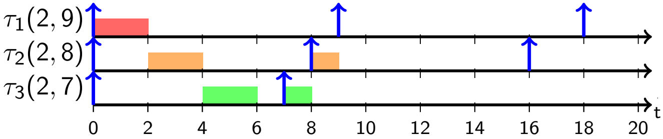


Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.

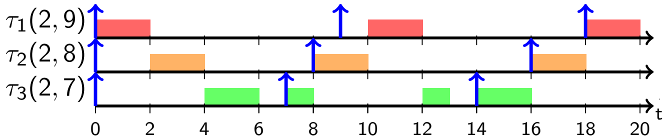


- Behaviour of Guptas and Doshis implementation of nested preemption in NXTOSEK (technical report in 2014).

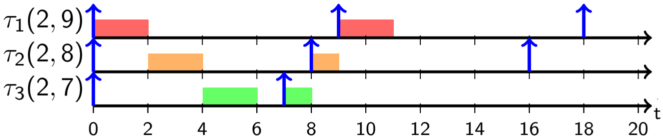


Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.

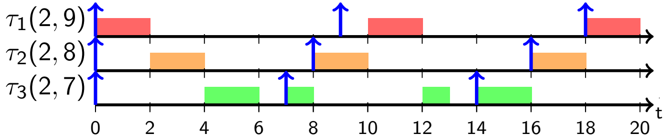


- Behaviour of Guptas and Doshis implementation of nested preemption in NXTOSEK (technical report in 2014).

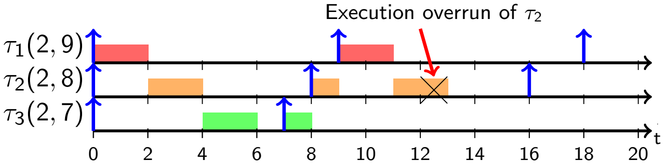


Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.

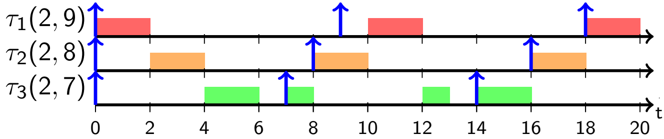


- Behaviour of Guptas and Doshis implementation of nested preemption in NXTOSEK (technical report in 2014).

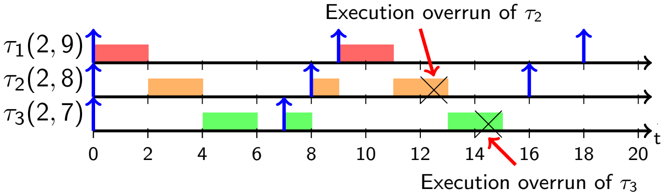


Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.

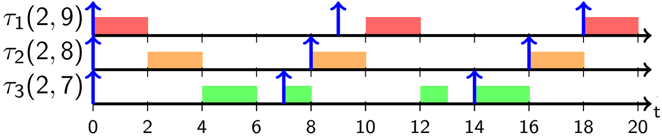


- Behaviour of Guptas and Doshis implementation of nested preemption in NXTOSEK (technical report in 2014).

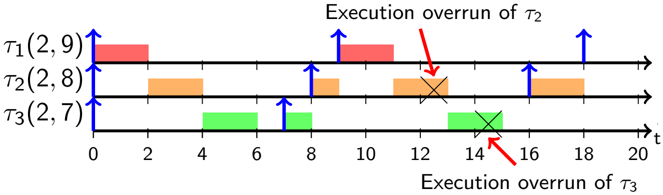


Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.

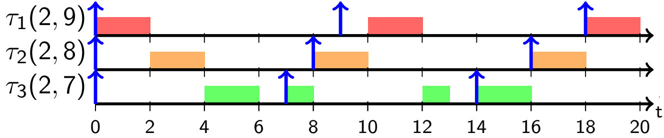


- Behaviour of Guptas and Doshis implementation of nested preemption in NXTOSEK (technical report in 2014).

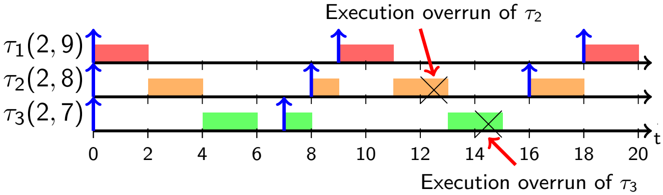


Issues in NXTOSEK

- NXTOSEK is not able to provide nested preemption.

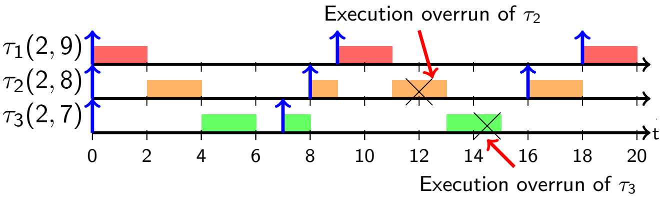


- Behaviour of Guptas and Doshis implementation of nested preemption in NXTOSEK (technical report in 2014).



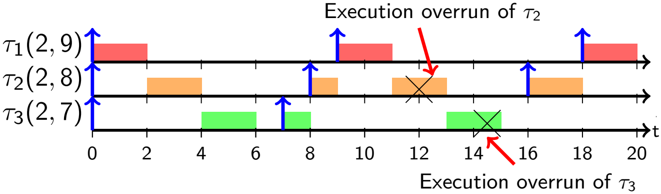
Issue in EV3OSEK

- The same issue as described by Gupta and Doshi was observed in EV3OSEK.



Issue in EV3OSEK

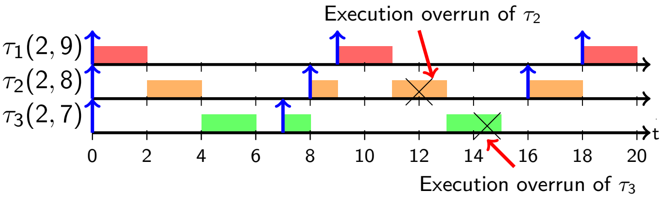
- The same issue as described by Gupta and Doshi was observed in EV3OSEK.



- Preempted jobs re-execute.

Issue in EV3OSEK

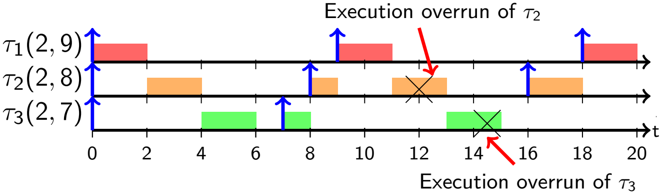
- The same issue as described by Gupta and Doshi was observed in EV3OSEK.



- Preempted jobs re-execute.
- Re-execution leads to longer WCET.

Issue in EV3OSEK

- The same issue as described by Gupta and Doshi was observed in EV3OSEK.



- Preempted jobs re-execute.
- Re-execution leads to longer WCET.
- Longer WCET leads to deadline misses.

Issues in EV3OSEK

- What is the cause of re-execution?

Issues in EV3OSEK

- What is the cause of re-execution?
    - Context switching has to be erroneous.

Issues in EV3OSEK

- What is the cause of re-execution?
  - Context switching has to be erroneous.
  - Handled by: IRQ-handler and the dispatch routines.

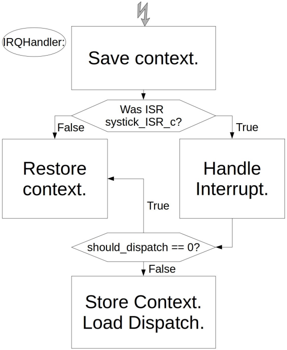
Issues in EV3OSEK

- What is the cause of re-execution?
  - Context switching has to be erroneous.
  - Handled by: IRQ-handler and the dispatch routines.
  - IRQ-Handler is a rewrite.

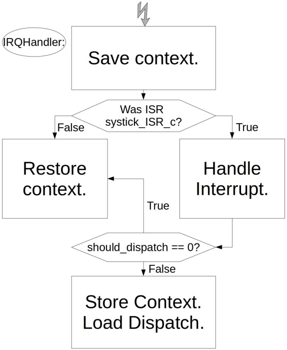
Issues in EV3OSEK

- What is the cause of re-execution?
  - Context switching has to be erroneous.
  - Handled by: IRQ-handler and the dispatch routines.
  - IRQ-Handler is a rewrite.
  - Dispatch routines are ported from NXTOSEK.

EV3OSEKs Interrupt Routines

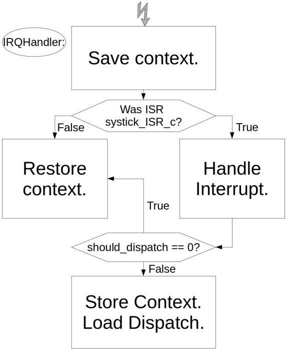


EV3OSEKs Interrupt Routines



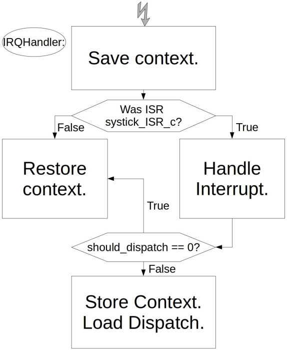
1. **Issue:** Status register is not part of context switching.

EV3OSEKs Interrupt Routines



1. **Issue:** Status register is not part of context switching.
2. **Issue:** The lookup register is overwritten.

EV3OSEKs Interrupt Routines



1. **Issue:** Status register is not part of context switching.
2. **Issue:** The lookup register is overwritten.
3. **Issue:** The dispatch routine is loaded with an offset of -4.

Fixing EV3OSEKs IRQ-Handler

- Status register has to be part of save context routines.

```
IRQHandler:
-  STMFD    r13!, {r0-r12, lr}     @ push r0-r12 and lr
```

Fixing EV3OSEKs IRQ-Handler

- Status register has to be part of save context routines.

```
IRQHandler:
+   SUB      lr, lr, #0x4          @ remove offset from lr
+   STMFD    r13!, {r0-r12}        @ push r0-r12
+   STMFD    r13!, {lr}            @ push lr
+   MRS      r0, spsr              @ load spsr to r0
+   STMFD    r13!, {r0}            @ push spsr
```

Fixing EV3OSEKs IRQ-Handler

- Status register has to be part of restore context routines.

```
SkipButtons:
-   LDMFD    r13!, {r0-r12, lr}   @ restore r0-r12 and lr
-   SUBS     pc, lr, #0x4         @ jump to address lr-4
```

Fixing EV3OSEKs IRQ-Handler

- Status register has to be part of restore context routines.

```
SkipButtons:
+   LDMFD    r13!, {r0}          @ pop spsr
+   MSRNE    spsr_cxsf, r0       @ restore spsr
+   LDMFD    r13!, {lr}          @ restore lr
+   LDMFD    r13!, {r0-r12}      @ restore r0-r12

+   MOVS     pc, lr              @ jump to address lr
```

Fixing EV3OSEKs IRQ-Handler

- The lookup register is overwritten and has to be pushed to system stack, before leaving interrupt-handler.
- The dispatch routine is loaded with an offset of -4.

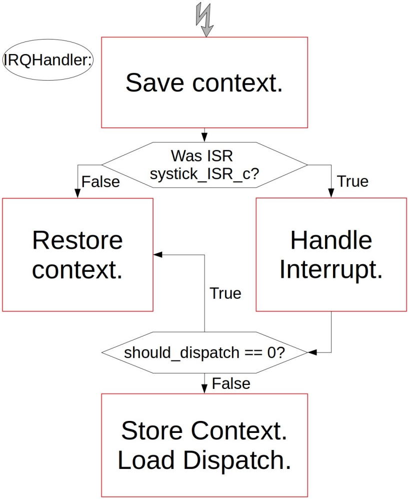
```
-   LDMFD    r13!, {r0-r12, lr}  @ pop r0-r12 and lr

-   LDR      lr, =dispatch       @ load jump address to lr
-   SUBS     pc, lr, #4          @ jump to address in lr
```

Fixing EV3OSEKs IRQ-Handler

- The lookup register is overwritten and has to be pushed to system stack, before leaving interrupt-handler.
- The dispatch routine is loaded with an offset of −4.

```
+  LDMFD    r13!, {r0}                @ pop spsr in r0
+  MSRNE    spsr_cxsf, r0             @ restore spsr
+  LDMFD    r13!, {r2}                @ pop lr in r2
+  MSR      cpsr_c, #MODE_SYS|I_F_BIT @ Change to System-mode
+  STMFD    r13!, {r2}                @ push lr on System-stack
+  MSR      cpsr_c, #MODE_IRQ|I_F_BIT @ Return to IRQ-mode

+  LDMFD    r13!, {r0-r12}            @ pop r0-r12

+  LDR      lr, =dispatch_irq         @ jump to dispatch
+  MOVS     pc, lr
```

Fixing EV3OSEKs IRQ-Handler



IRQHandler:

Save context.

Was ISR systick_ISR_c?

False

True

Restore context.

Handle Interrupt.

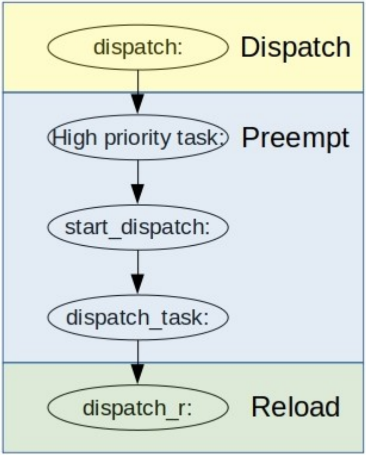
True

should_dispatch == 0?

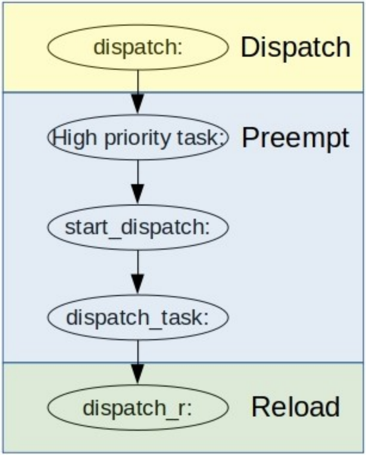
False

Store Context. Load Dispatch.

EV3OSEKs Dispatch Routines

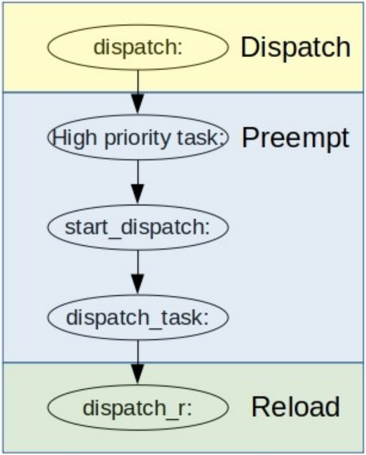


EV3OSEKs Dispatch Routines



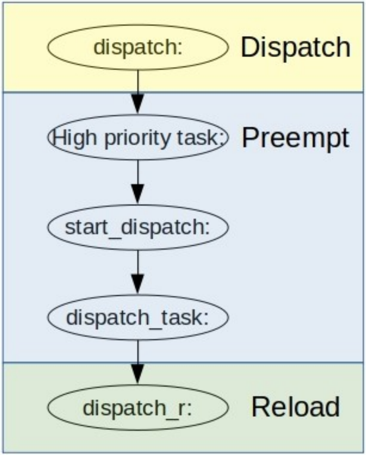
- **Dispatch:** Current task is saved and relinquishes CPU.

EV3OSEKs Dispatch Routines



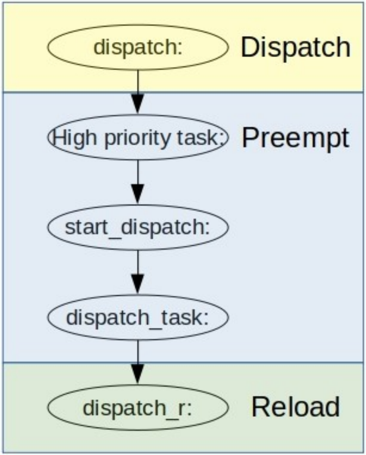
- **Dispatch:** Current task is saved and relinquishes CPU.
- **Preempt:** Higher priority task is loaded and finishes execution.

EV3OSEKs Dispatch Routines



- **Dispatch:** Current task is saved and relinquishes CPU.
- **Preempt:** Higher priority task is loaded and finishes execution.
- **Reload:** Preempted task is restored to CPU.

EV3OSEKs Dispatch Routines



- **Dispatch:** Current task is saved and relinquishes CPU.
- **Preempt:** Higher priority task is loaded and finishes execution.
- **Reload:** Preempted task is restored to CPU.

- **Issue:** The status register is neither saved, nor restored.

Fixing EV3OSEKs Dispatching Routines

- The status register is neither saved, nor restored.

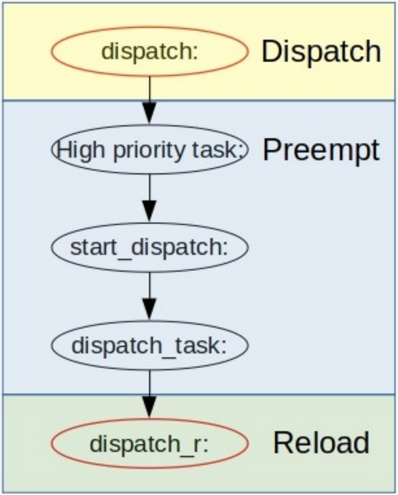
```
dispatch:
-  stmfd sp!, {r0-r12, lr} @ push r0-r12 and lr

dispatcher_r:
   BL      IntMasterIRQEnable  @ enable interrubts
   BL      IntMasterFIQEnable
   ldmfd sp!, {r0-r12}         @ restore r0-r12
   ldmfd sp!, {lr}             @ restore spsr value in lr
   mov   pc, lr                @ jump to lr
```

Fixing EV3OSEKs Dispatching Routines

- The status register is neither saved, nor restored.

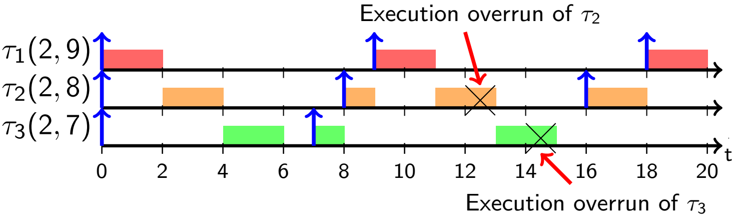
```
dispatch:
+  stmfd sp!, {lr}         @ push lr
+dispatch_irq:             @ lr already pushed by IRQ-Handler
+  mrs     lr, spsr        @ save spsr in lr
+  stmfd sp!, {lr}         @ push spsr
+  stmfd sp!, {r0-r12}     @ push r0-r12

dispatcher_r:
   BL      IntMasterIRQEnable  @ enable interrubts
   BL      IntMasterFIQEnable
   ldmfd sp!, {r0-r12}         @ restore r0-r12
   ldmfd sp!, {lr}             @ restore spsr value in lr
+  msrne   spsr_cxsf, lr       @ write lr in spsr
+  ldmfd sp!, {lr}^            @ restore lr and write spsr to cpsr
   mov   pc, lr                @ jump to lr
```

Fixing EV3OSEKs Dispatching Routines

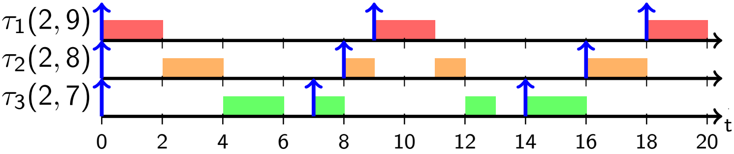


Evaluation

- Old dispatching routine of EV3OSEK:



- Corrected dispatching routine of EV3OSEK:



Evaluation

- Example to verify the functional correctness of the dispatcher.

Evaluation

- Example to verify the functional correctness of the dispatcher.
- The tasks are specified as:
  - $\tau_1 = (2, 5)$
  - $\tau_2 = (2, 8)$
  - $\tau_3 = (2, 10)$

Evaluation

- Example to verify the functional correctness of the dispatcher.
- The tasks are specified as:
  - $\tau_1 = (2, 5)$
  - $\tau_2 = (2, 8)$
  - $\tau_3 = (2, 10)$
- These tasks have a hyper-period of 40 seconds.

Evaluation

- Example to verify the functional correctness of the dispatcher.
- The tasks are specified as:
  - $\tau_1 = (2, 5)$
  - $\tau_2 = (2, 8)$
  - $\tau_3 = (2, 10)$
- These tasks have a hyper-period of 40 seconds.
- This equation determines how many jobs are on average released in the intervals of length $t = 10$ seconds.

$$\begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} = \begin{pmatrix} \frac{8}{4} \times t \\ \frac{5}{4} \times t \\ \frac{4}{4} \times t \end{pmatrix} = \begin{pmatrix} 2t \\ 1.25t \\ t \end{pmatrix}$$

Evaluation

- To verify that the stack and the lookup register are handled correctly, the example is executed for 600 seconds (10 min).

$$t = 600001(ms) \approx 60 \times 10sec \Rightarrow \begin{pmatrix} l_1(60) = 120 \\ l_2(60) = 75 \\ l_3(60) = 60 \end{pmatrix}$$

Evaluation

$$t = 600001(ms) \approx 60 \times 10sec \Rightarrow \begin{pmatrix} l_1(60) = 120 \\ l_2(60) = 75 \\ l_3(60) = 60 \end{pmatrix}$$

Task $i(l_1(t), l_2(t), l_3(t))$ start at $t \times 10^4$.

```
Task 1(0, 0, 0) start at 1.
Task 1(1, 0, 0) end at 2005.
Task 2(1, 0, 0) start at 2008.
Task 2(1, 1, 0) end at 4003.
Task 3(1, 1, 0) start at 4005.
Task 1(1, 1, 1) start at 5001.
Task 1(2, 1, 1) end at 6995.
...
Task 1(119, 75, 60) start at 595001.
Task 1(120, 75, 60) end at 596996.
Task 3(120, 75, 60) end at 597992.
Task 1(120, 75, 60) start at 600001.
```

Conclusion

- EV3OSEKs faults have been analysed.
- Multitasking is enabled in EV3OSEK.
- The proposed implementation works.

Current State of EV3OSEK

Analog devices	Status	Comment
Lego EV3 touch sensor	✓	Is not implemented with interrupt.
Lego EV3 sound sensor	*n.t.*	Implemented but not tested.
Lego EV3 light sensor	∼	Not working always responses with 0.
Lego NXT light sensor	✓	Works.

Digital devices	Status	Comment
Lego EV3 ultrasonic sensor	∼	Every third API call returns 0.
Lego EV3 gyro sensor	*X*	Not implemented.
HiTechnic gyro sensor	✓	Works.
HiTechnic acceleration sensor	*n.t.*	Implemented but not tested.
HiTechnic compass	*n.t.*	Implemented but not tested.
HiTechnic color sensor V2	*n.t.*	Implemented but not tested.

Current State of EV3OSEK

Analog devices	Status	Comment
Lego EV3 touch sensor	✓	Is not implemented with interrupt.
Lego EV3 sound sensor	*n.t.*	Implemented but not tested.
Lego EV3 light sensor	∼	Not working always responses with 0.
Lego NXT light sensor	✓	Works.

Digital devices	Status	Comment
Lego EV3 ultrasonic sensor	∼	Every third API call returns 0.
Lego EV3 gyro sensor	*X*	Not implemented.
HiTechnic gyro sensor	✓	Works.
HiTechnic acceleration sensor	*n.t.*	Implemented but not tested.
HiTechnic compass	*n.t.*	Implemented but not tested.
HiTechnic color sensor V2	*n.t.*	Implemented but not tested.

- LED, Bluetooth, SD-Card and EV3-Motors not working.

Current State of EV3OSEK

Analog devices	Status	Comment
Lego EV3 touch sensor	✓	Is not implemented with interrupt.
Lego EV3 sound sensor	*n.t.*	Implemented but not tested.
Lego EV3 light sensor	$\sim$	Not working always responses with 0.
Lego NXT light sensor	✓	Works.

Digital devices	Status	Comment
Lego EV3 ultrasonic sensor	$\sim$	Every third API call returns 0.
Lego EV3 gyro sensor	X	Not implemented.
HiTechnic gyro sensor	✓	Works.
HiTechnic acceleration sensor	*n.t.*	Implemented but not tested.
HiTechnic compass	*n.t.*	Implemented but not tested.
HiTechnic color sensor V2	*n.t.*	Implemented but not tested.

- LED, Bluetooth, SD-Card and EV3-Motors not working.

Thank You!